2 - Pointers

3 - Pair , 4 - P

K - P

- ⇒ Physical significance of pointer. ⟺ (i , j) / [i , j]
- ⇒ How to initialize the pointer (why?)
- ⇒ How to move the pointer (reason)
- ⇒ When to stop.

---

(1) Given a **Sorted array** with distinct integer. (+ve)

K

find → pair sum

$a[i] + a[j] = K$ $\boxed{(i != j)}$

1, 4, 5, 6, 11

K = 10

(4, 6)

TC → O(N)
SC → O(1)

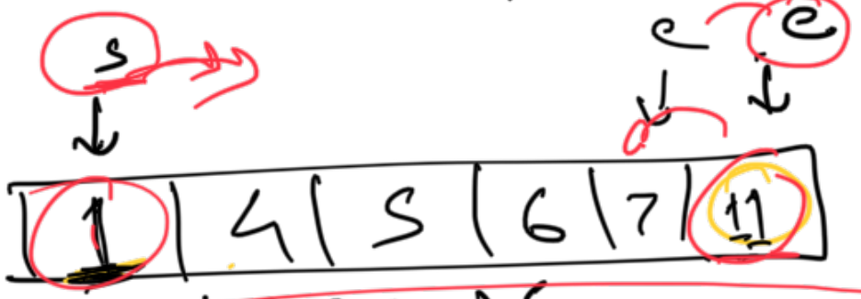
O(1) Space ε

Brute force

2 loops, check every pair

i → i
i → j

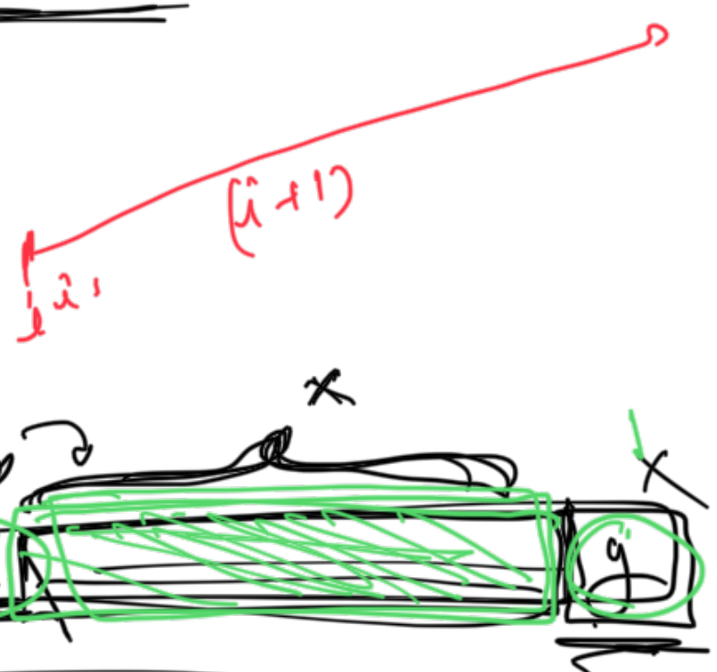
$K = 10$



$O(N^2)$

```
while(s<e){
  if (a[s] + a[e]  > K) {
      e--;
  }
  else if( a[s] + a[e] < K){
      s++;
  }
  if ( a[s] + a[e] == K)
      ret (s, e);
}
```

why? C++.



$x+y > K$	→ y is useless
$x+y < K$	→ x is useless

$x + y = K$

$x + (<y) < K$

With repetition

K = 10

Count pairs with sum

cnt = 2

$\frac{2 \times 2}{2} \rightarrow 4$ (i, j)

q(s) = 3

q(e) = 3 $\Rightarrow$ 9 ✗

$^{n}C_{2}$

3, 3, 3, 3, 3, 3 K = 6

s e

q(s) = 3

q(e) = 3 $\Rightarrow$ 9 ✗

$O(1)$

$\binom{n}{2}$

Q2 Diff K

Facebook

Intuit Bits

Sorted array. +ve int

$\Rightarrow$ K (+ive)

i, j

$(i != j)$

$A[i] - A[j] = K$

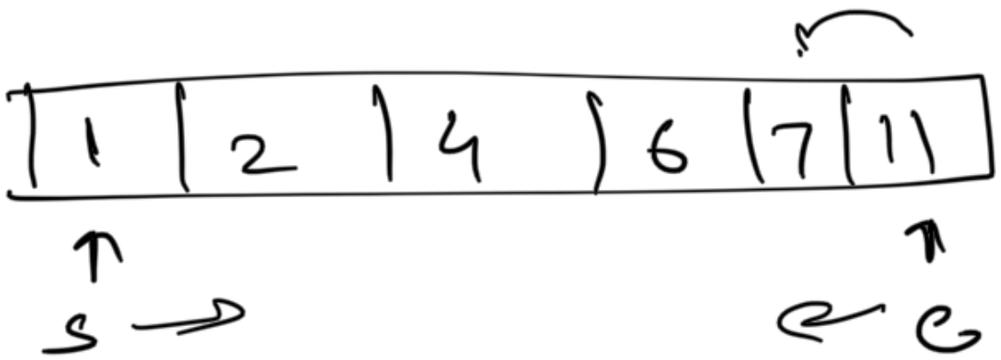
$A[i] = A[j] + K$

$O(1)$ space

$O(N^2)$

$O(N \log N)$

Brute force ⇒ $O(N^2)$



$a[e] - a[s] > k$

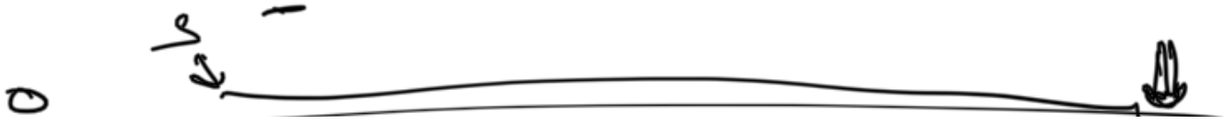
s+1

e--

⟹ same effect ?

$a[e] - a[s] <= k$ ✗

$x - y = z$

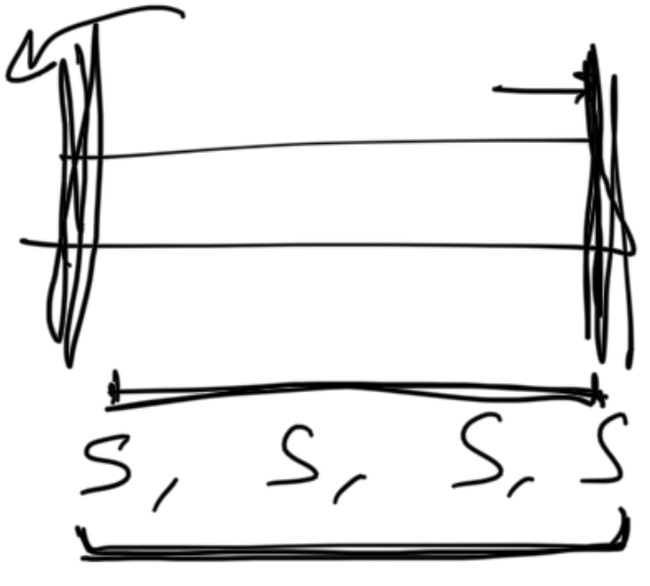


s → ∫ dx

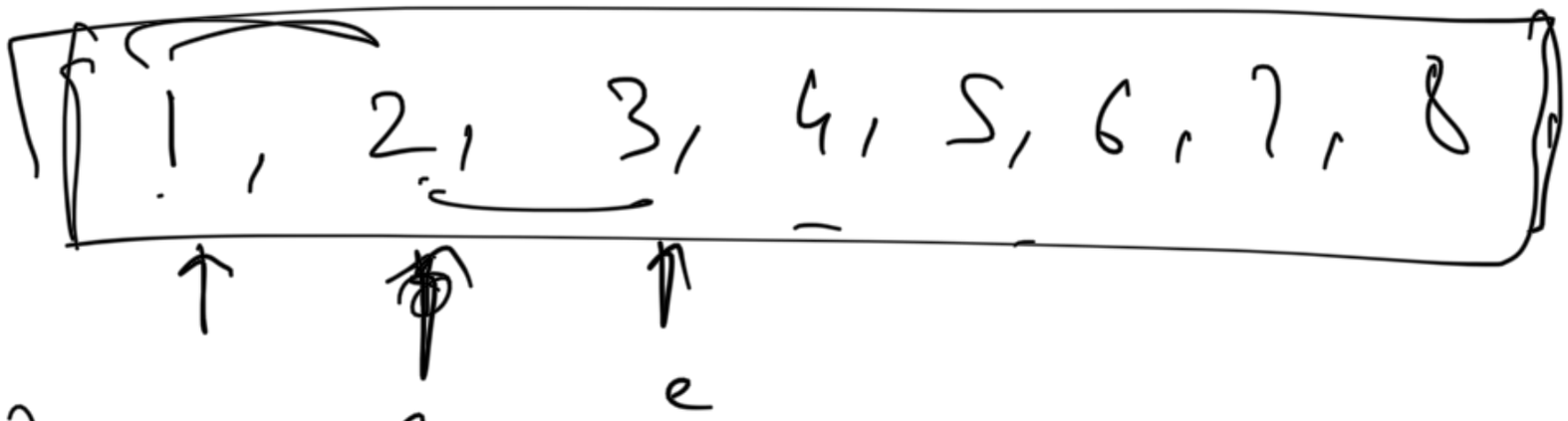
e → ∫ diff

diff = a(e) − a(s)

```
if (diff >= k)   s++;
else if (diff < k)   e++;
else if (diff == k)   result(i, j);
```



k=0 1, 2, S, S, S, S, 7, 8



k=1

(1,2) (3,4)

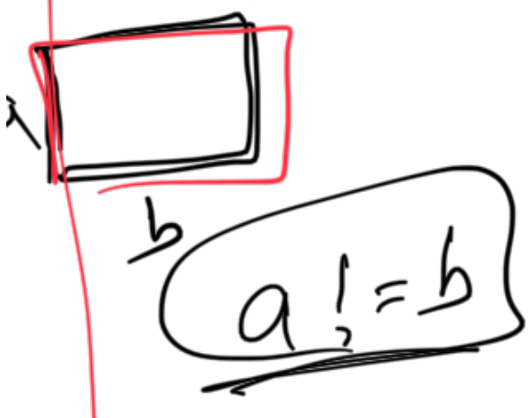
Q.3 Amazon

sorted array → (distinct elem)

[2, 3, 5]

Count no. of rectangles with distinct config.

(a x b) != (b x a)



a != b

Area of rect < B

[2, 3, 5]

B = 15

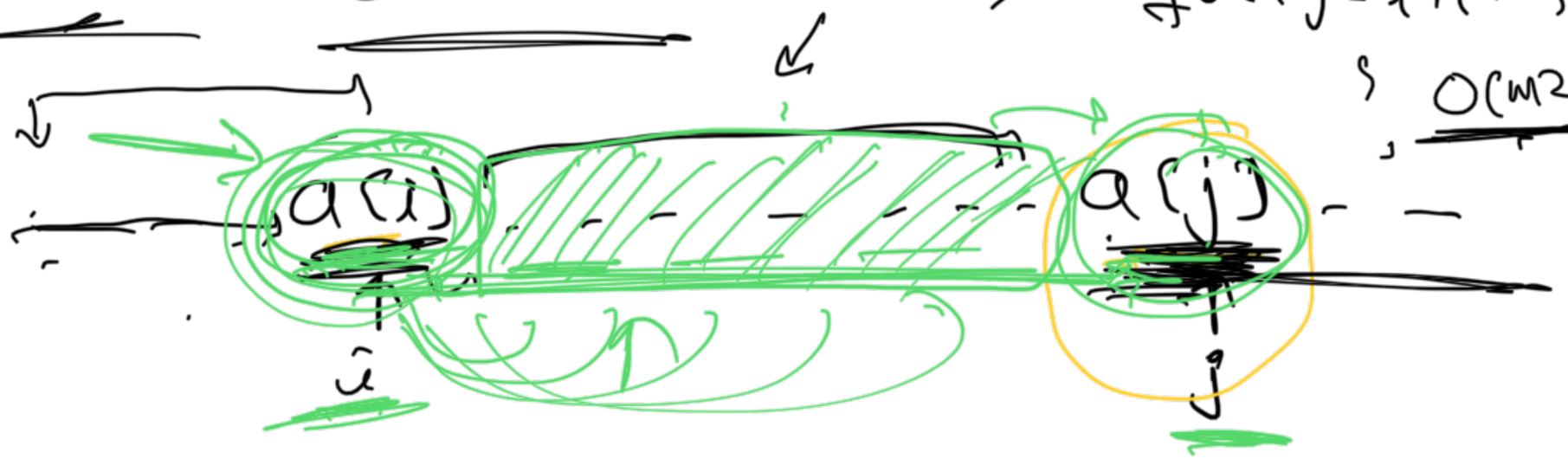
4

2 x 2	3 x 2	5 x 2
2 x 3	3 x 3	~~8~~
2 x 5	~~3 x 5~~	

G

Brute force $O(N^2)$

for (i = 0 → n)
  for (j = i+1 → n)
    { $O(M^2)$



if (a[i] x a[j] > B)

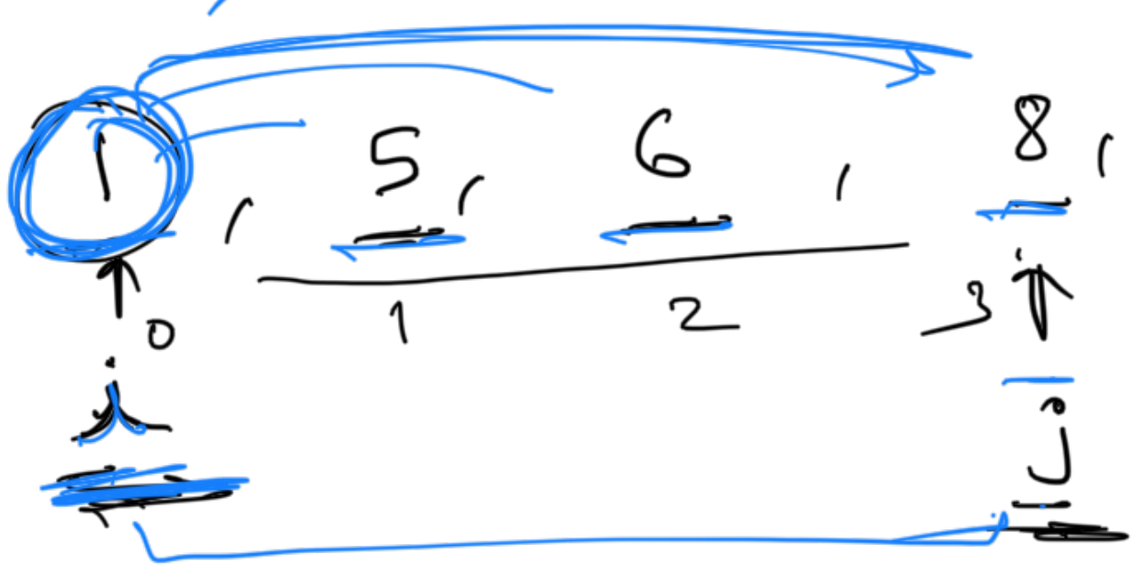
a(i), a(j)

else if (a[i] × a[i] < B) { a(i), a(j)
a(j), a(i)

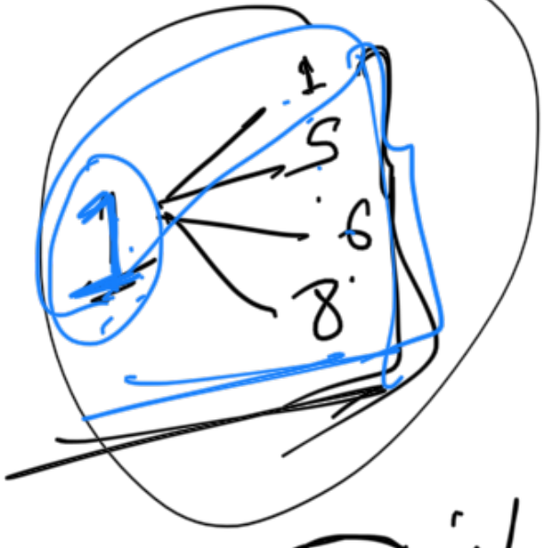
ans += 2 * (j - i + 1) - 1 ;

i + 1 ;

(2 × l - 1)



B = 9



if (a[i] × a[j] < 9)

(1 × 1) ((1 × 1))

(1 × 5) (5 × 1)

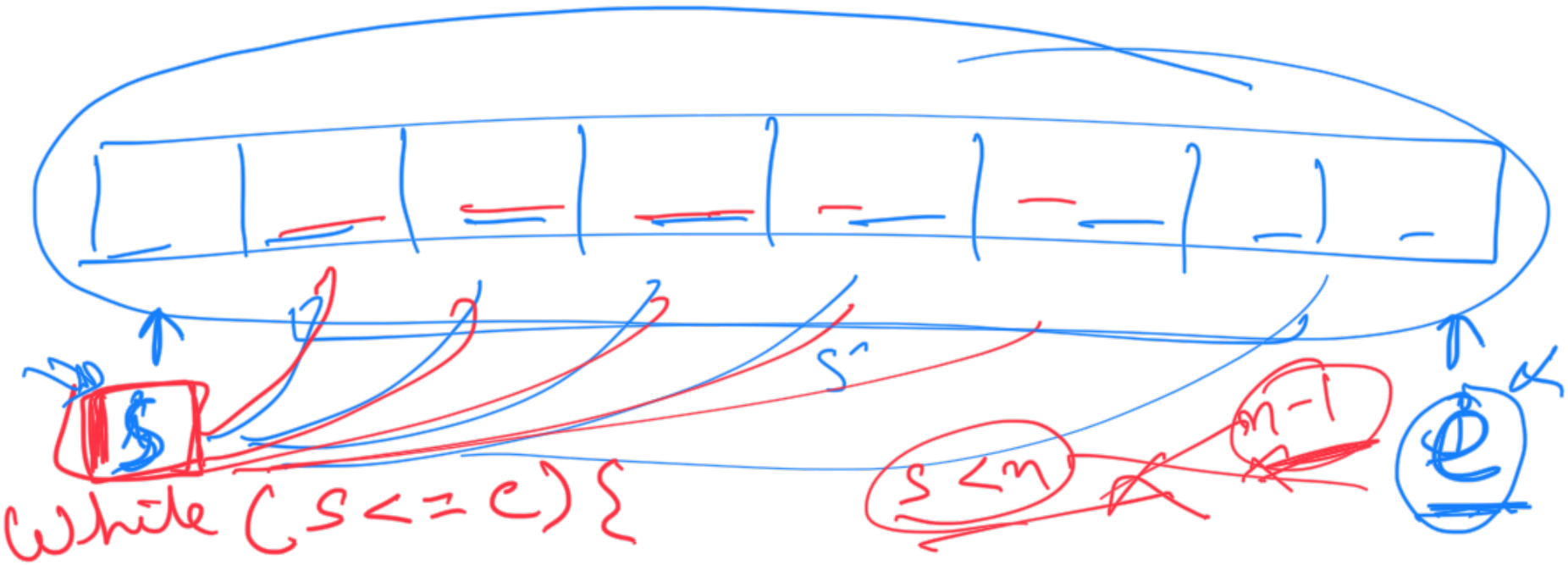
(1 × 6) (6 × 1)

ans += 2 × (j - i + 1) - 1]

(1x6) (8^1)

i++;

Sum=10



S.

while (s<=e){

s<e

if (a[s] * a[e] >= B) {

e--;

}

else if (a[s] * a[e] < B) {

O(N)

ans = ans + (e - s + 1) x 2 - 1 ;

s + + ;

cont C_2

B = 10

1

1 x 1

TB

⇒ If there are duplicates ?

HM

Q4 Amazon SDE 2

A : { 1, 4, 5, 8, 10 }

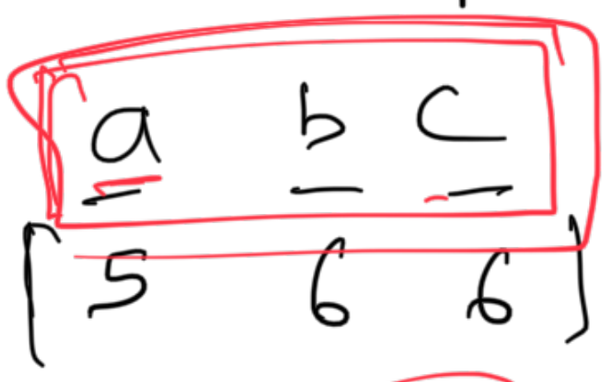
B : { 6, 9, 15 }

} 3 Sorted array

c : { 2, 3, 6, 6 }

MINIMIZE

| max(a, b, c) − min(a, b, c) |



max(a, b, c) → 6
min(a, b, c) → 5

] → 1

Brute force

O(N³)

for (i = 0 → A.len) {

i

1, 4, 5, 8, 10 — 18

6, 9, 18

2, 3, 6, 6, 12

min = 6

min = 1 2 3 ⇒ 5

(6, 1, 12)

for (j=0 → B.h) {

for (k=0 → C.h

[a(i), b(j), c(k)]

18

man(a, b, c) ← min(a, b, c)

=0

if I have to minimize

→ reduce : X Not Possible

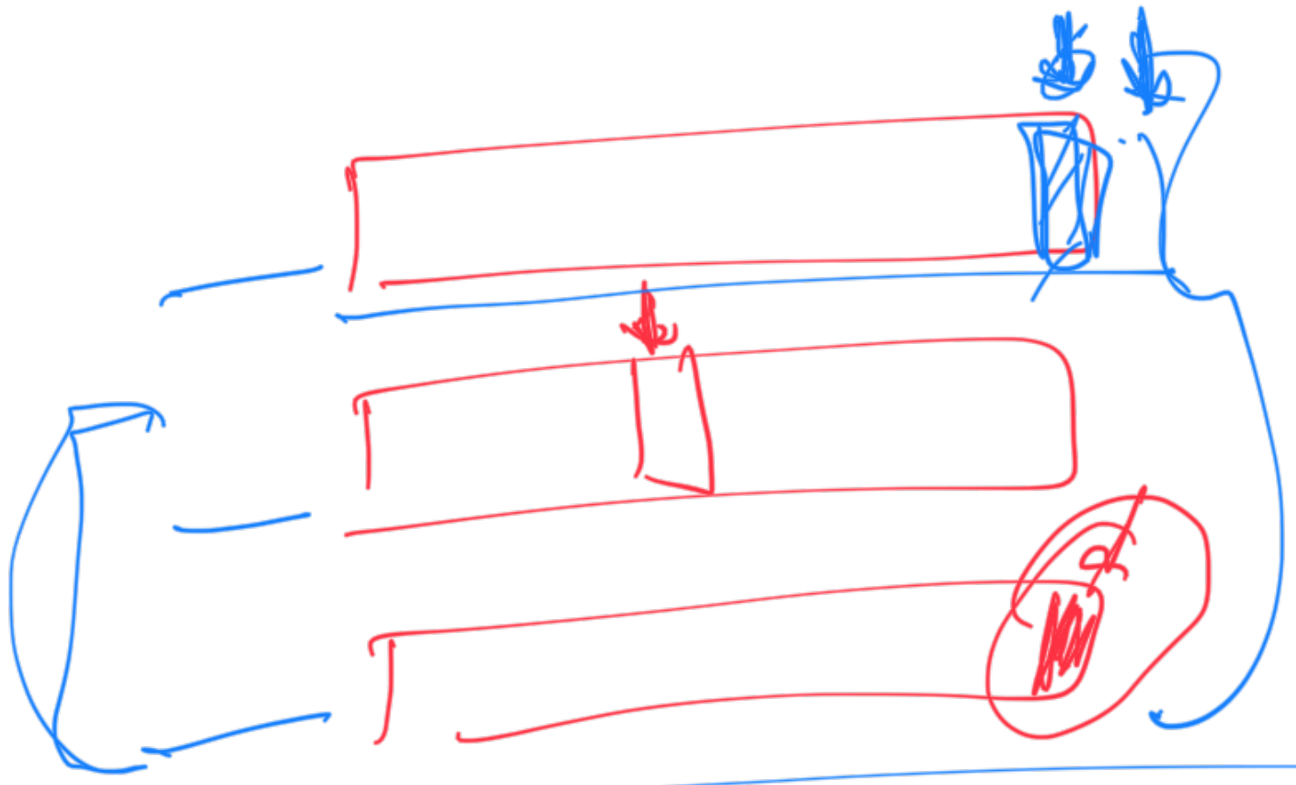
→ return y

a(i) b(j) c(k)

1 6 2

min = 1 , max = 6

4 6 2 $m_i = 2$



i 8 10

j 6 100

k 6 200 1

min i j k

8 1 6 1 6

mi = 6 ma = 8

2 8 16100, 200

mi = 8 m = 200

Q5

A. [N]

+ve

Unsorted

1 week before with

Amazon

Microsoft

1, 4, 3, 10, 5, 20

return T if there exist a subarray with sum K

1, 4 = 153, → 10 T 5, 20

Const

Final

j = 0 → n

P → | 1 | 5 | 8 | 18 | 23 | 43 |

i

j

i^th → Σ [0, i]

Sum [l, r] = P[r] - P[l-1]

= P[i] - P[j]

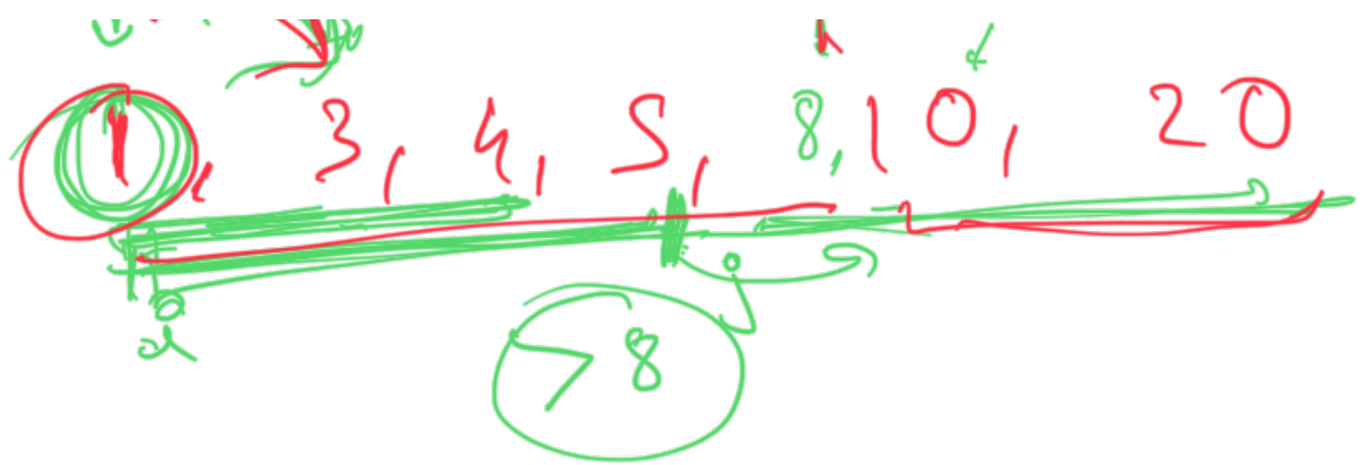
K

Diff

Hash

TC → O(n)

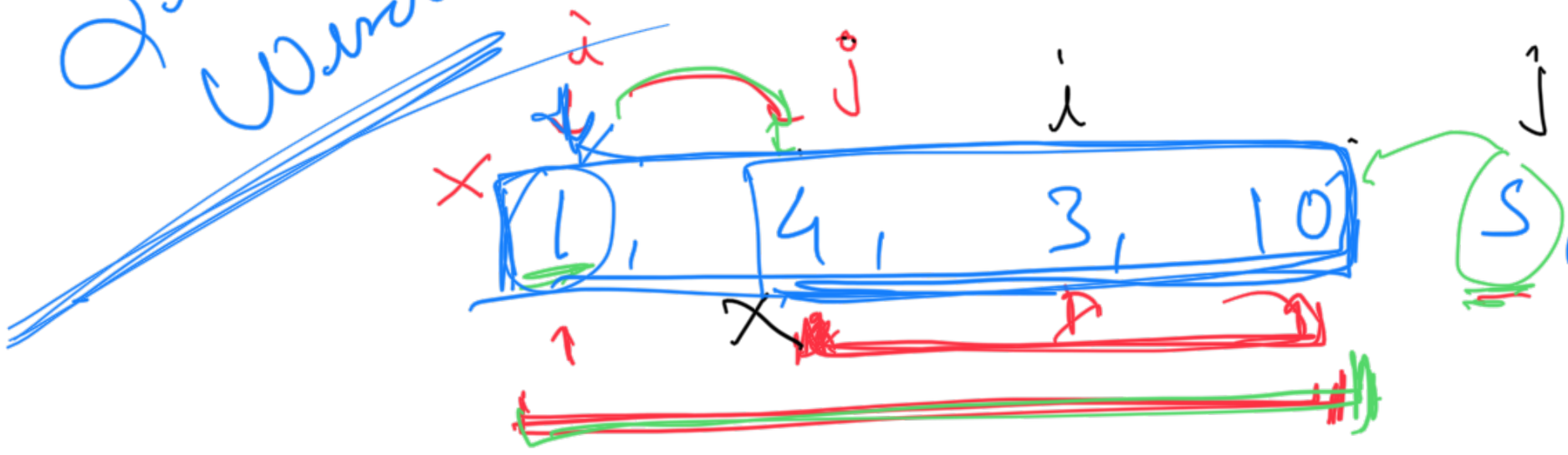
SC → O(n)

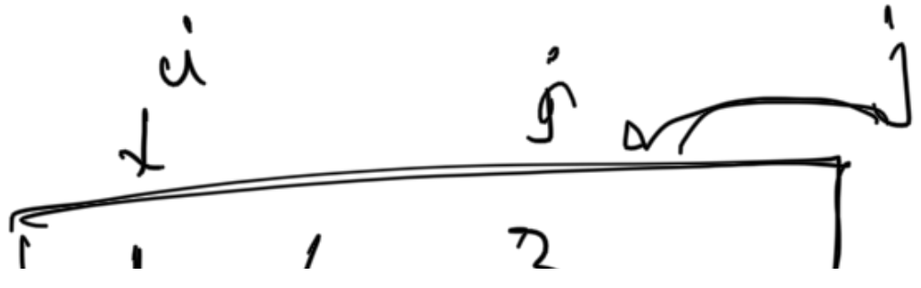


$k = 8$

Sliding window

sum (i - j) > 8





1, 4, 21 | 10

3 | 1, 4, 10, 5, 20

K = 15

Sum = 3 → 4 → 8 → 18 → 15

O(N)

7 K | O

6 | 20 10 | 22 K=15 | 20 = 19 - 29

i j

Q Amazon → Microsoft

Heights

[1, 8, 6, 2, 5, 4, 8, 3, 7]

+

Adobe

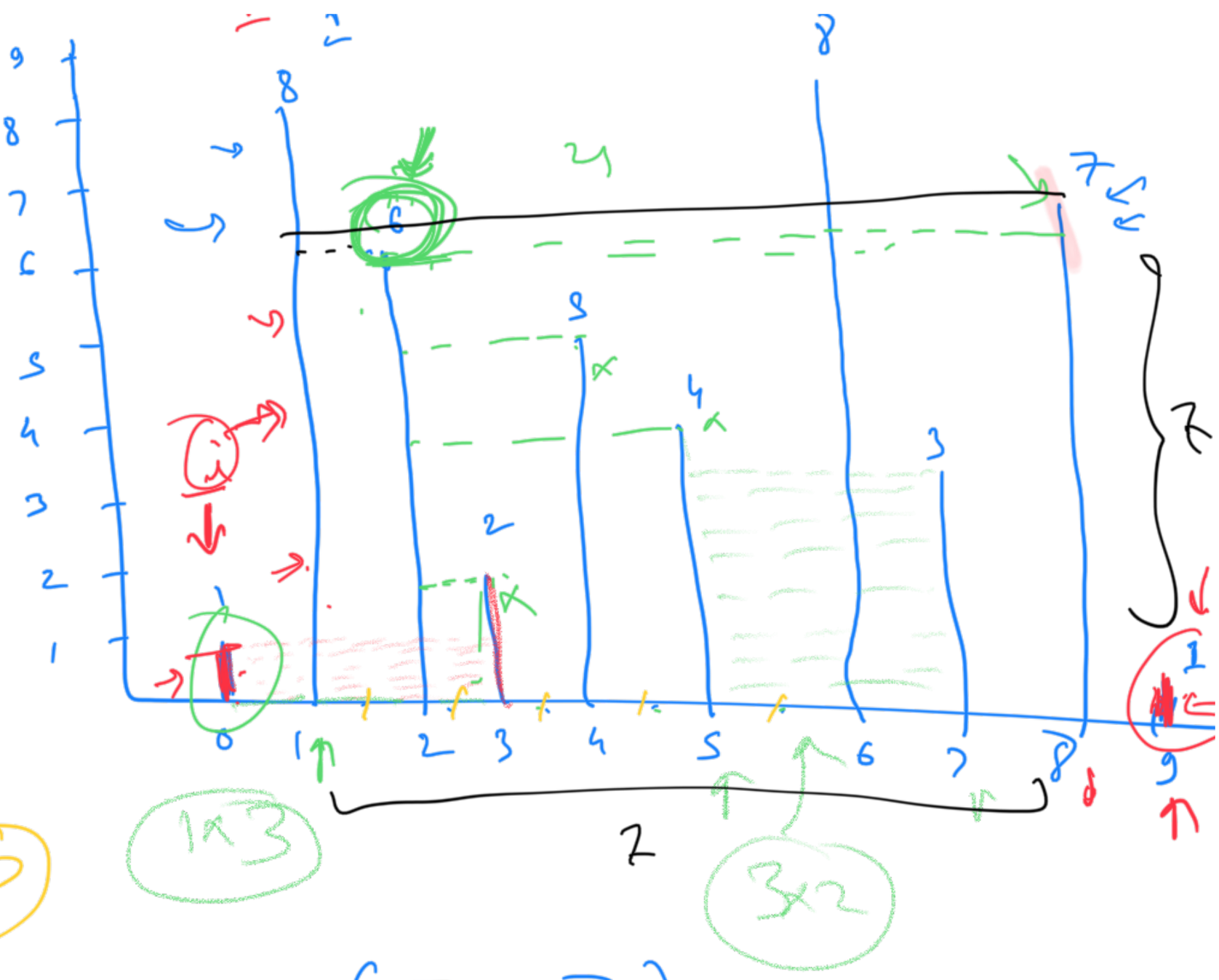
+

Flipkart

+

Facebook

Contains either most evatr



8x5 = 40

Ans (7 x 7) = 49

Brute force

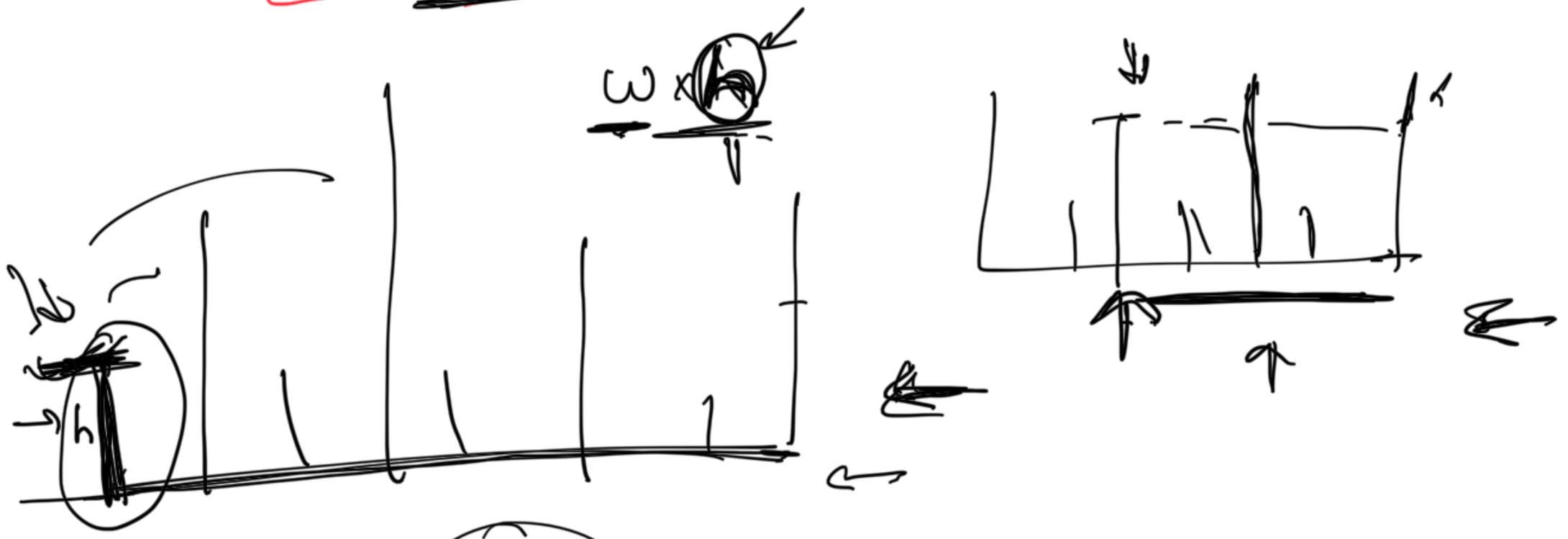
O(N²)

Observation

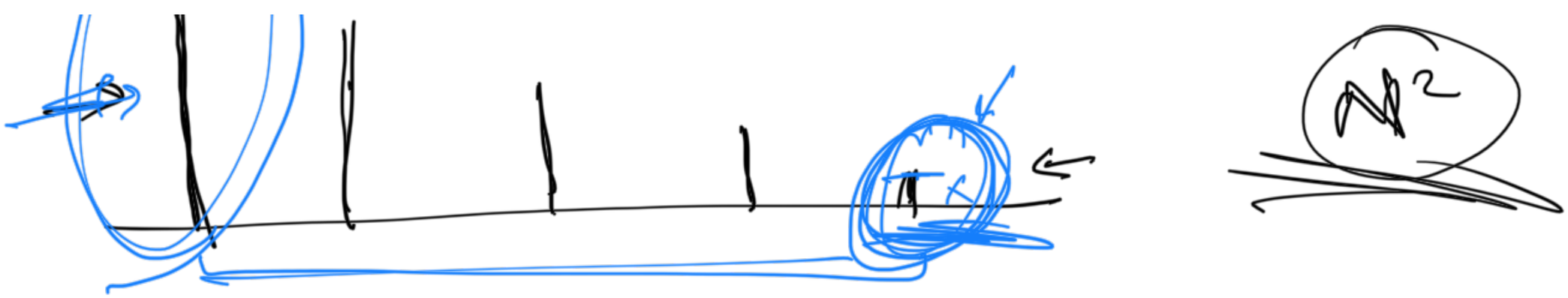
The max area for any $h(i)$

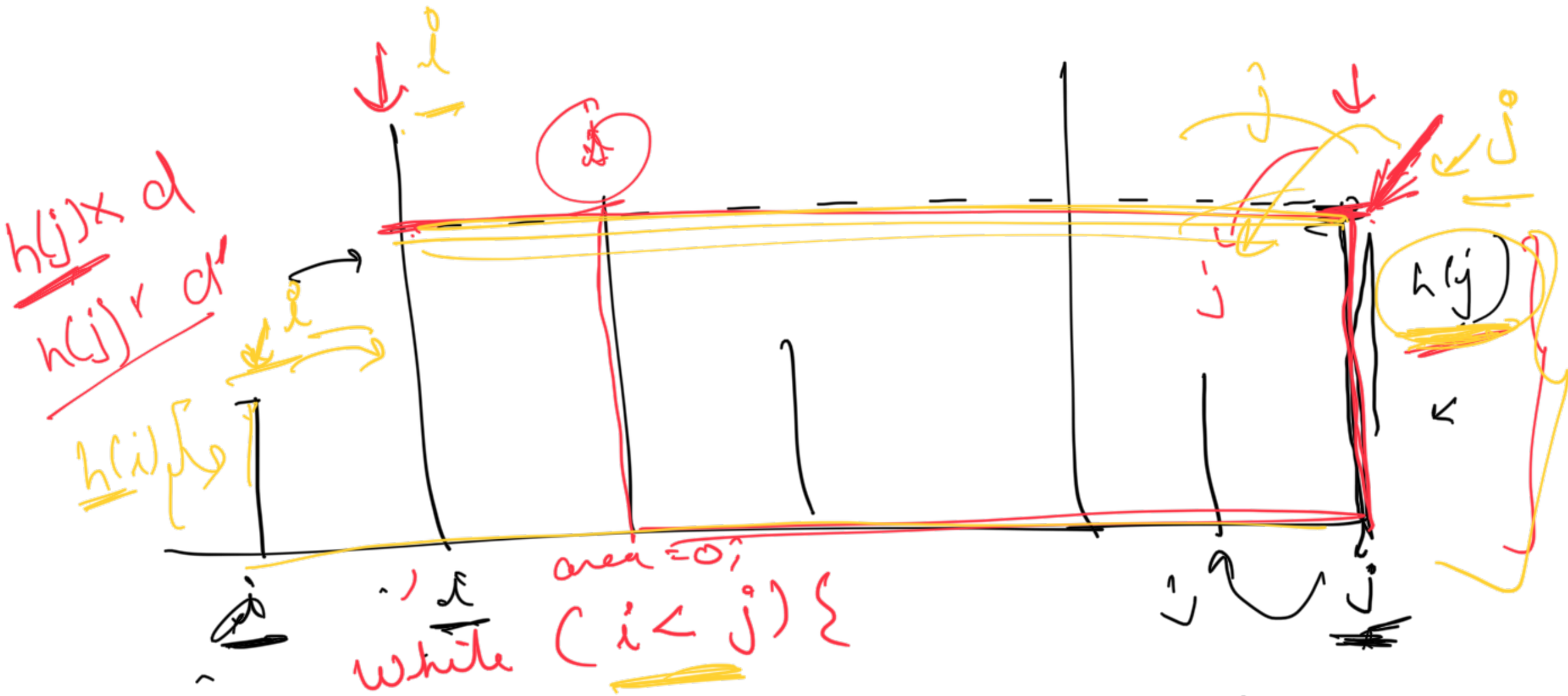
comes from the farthest $h(j)$

s.t $h(j) >= h(i)$



TC $\Rightarrow$ N^2





if (h[j] >= h(i)) {

area = (i+1);

}

else { area = max(h[j]*(j-i), area));

j--;

}

}

nC2

[i, j]*2 -1